

基于 Renyi 熵的 SDN 自主防护系统

赵普¹, 赵文涛¹, 付章杰², 刘强¹

(1. 国防科技大学计算机学院, 湖南 长沙 410073;

2. 南京信息工程大学计算机与软件学院, 江苏 南京 210044)

摘要: 针对 SDN 架构下的常见网络异常行为, 提出了一套基于 Renyi 熵的 SDN 自主防护系统, 该系统可实现网络异常行为检测、诊断及防御。系统无须引入第三方测量设备, 直接利用 OpenFlow 交换机流表信息。首先, 通过计算和检测特征熵值, 实现异常网络行为的检测。然后, 进一步分析 OpenFlow 流表信息, 实现异常行为的诊断。最后, 实施防御控制措施, 建立一套黑名单机制, 将产生异常行为的主机加入黑名单, 并阻塞相应的异常流量。为了验证系统的有效性, 在 Floodlight 控制器上开发了原型。Mininet 上的仿真实验表明, 系统能够有效检测、诊断及防御网络中常见的异常行为, 且具有较低的部署成本, 增强了 SDN 的安全性。

关键词: 软件定义网络; 网络异常检测; Renyi 熵; OpenFlow 协议

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2021049

SDN self-protection system based on Renyi entropy

ZHAO Pu¹, ZHAO Wentao¹, FU Zhangjie², LIU Qiang¹

1. College of Computer, National University of Defense Technology, Changsha 410073, China

2. School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

Abstract: Aiming at the abnormal behaviors in SDN architecture, a self-protection system based on Renyi entropy that implemented a set of detection, diagnosis and defense method of SDN abnormal behaviors was proposed. The system did not need to introduce the third-party measurement equipment, and directly used the flow table information of OpenFlow switches. Firstly, the abnormal network behavior was detected by calculating the characteristic entropy. Then, the information of the OpenFlow flow table was further analyzed to realize the diagnosis of abnormal behavior. Finally, a blacklist mechanism was established. And the system added the hosts with abnormal behavior to the blacklist and blocked the corresponding abnormal traffic. In order to verify the effectiveness of the system, a prototype was developed on the Floodlight controller. The simulation results on Mininet show that the system can effectively detect, diagnose and defend the abnormal behaviors. The system has low deployment cost, which enhances the security of SDN.

Keywords: software defined network, anomaly detection, Renyi entropy, OpenFlow protocol

收稿日期: 2020-09-16; 修回日期: 2020-11-16

通信作者: 刘强, qiangliu06@nudt.edu.cn

基金项目: 国家自然科学基金 (U1811462, 61702539); 湖南省自然科学基金 (2018JJ3611)

Foundation Items: The National Natural Science Foundation of China (U1811462, 61702539), The Natural Science Foundation of Hunan Province (2018JJ3611)

论文引用格式: 赵普, 赵文涛, 付章杰, 等. 基于 Renyi 熵的 SDN 自主防护系统[J]. 网络与信息安全学报, 2021, 7(3): 85-94.

ZHAO P, ZHAO W T, FU Z J, et al. SDN self-protection system based on Renyi entropy[J]. Chinese Journal of Network and Information Security, 2021, 7(3): 85-94.

1 引言

软件定义网络 (SDN, software defined network) 是 5G 网络和未来 6G 网络的重要使能技术之一, 它解耦了网络控制平面与数据平面, 为用户提供了灵活的编程接口以控制网络行为, 极大地增强了网络的灵活性。然而, 无论是 SDN 还是传统网络都易遭受异常流量的破坏, 如分布式拒绝服务 (DDoS, distributed denial of service) 攻击^[1-2], 异常流量会降低网络安全性和可用性。传统计算机网络中要实现异常检测需要部署专用设备, 由于传统网络分布式的特点, 这些专用设备的部署开销较大, 且降低了网络的灵活性。相比于传统网络, SDN 允许构建更可靠、可扩展和可管理的网络, 以便更容易地部署新的网络功能^[3]。因此, 网络异常检测及防御可以集中在 SDN 控制器上实现。然而, 基于 SDN 的网络本身并没有异常检测及防御功能, 存在安全隐患。尽管如此, SDN 范式可为网络异常检测和防御系统设计提供新“抓手”, 原因如下: SDN 提供了网络全局视图; SDN 非常便于收集流量统计信息; SDN 利用集中式控制平面, 可方便用户部署新的应用^[4]。

本文旨在实现一套面向 SDN 的自主防护系统, 该系统联合执行网络异常流量的检测、诊断及防御等任务。同时, 该系统具有较强的可扩展性, 能够灵活地检测不同类型的网络异常行为。为此, 本文提出了一种基于 Renyi 熵的 SDN 自主防护系统。具体来讲, 系统周期性地提取 OpenFlow 交换机流表信息, 随后分别计算出流表项中不同项目的 Renyi 熵值。一旦发现熵值发生突变, 立即调用网络诊断模块来识别异常流量, 进而追查产生异常流量的网络节点, 最后形成 JSON(JS 对象简谱, JavaScript object notation) 格式的诊断报告, 并将诊断报告提交给网络防御模块。网络防御模块采取了黑名单机制, 将产生异常行为的主机加入黑名单, 以备进一步检测, 同时阻塞网络异常流量。具体防御措施可根据用户需要灵活拓展, 以增强网络的灵活性和安全性。

本文工作的主要贡献包括: ① 直接利用 OpenFlow 流表信息作为检测源数据, 不需要其他

网络检测开销; ② 针对一个 OpenFlow 控制域内小流量特点, 采用基于 Renyi 熵的检测方法, 有利于增大不同分布之间的差异, 相对于传统熵检测, 精度更高; ③ 采用模块化方式开发了原型系统, 各模块之间通过 JSON 标准格式数据沟通, 有利于后续功能拓展。为了进行概念性证明, 本文在 Floodlight 控制器上实现了原型系统, 采用 Mininet 仿真工具构建网络拓扑。此外, 本文采用 Scapy 工具产生业务流量和异常流量, 模拟了一个僵尸网络 (Botnet) 发动 DDoS 攻击的 3 个阶段 (嗅探、传播、攻击), 3 个阶段对应的异常行为分别是端口扫描、蠕虫病毒传播、发动 DDoS 攻击的网络流量。实验结果表明, 本文提出的系统可以有效地检测网络异常行为, 诊断异常行为类型和恶意行为产生源头, 并采取相应的防御措施来提升网络安全性。

2 SDN 安全研究现状

SDN 安全研究主要涉及两方面。

一方面, 研究者关注 SDN 范式本身的安全问题。由于 SDN 集中控制的特点, 它在控制平面、应用平面、数据平面和标准接口等方面面临一系列安全挑战, 如恶意应用、网络监听、IP 地址欺骗、DoS/DDoS 攻击和病毒木马攻击等, 这是制约 SDN 商用化的一个巨大阻碍。

另一方面, 用 SDN 思想解决传统网络面对的安全问题。由于 SDN 具有网络集中控制和流量的细粒度管理能力, 研究者倾向利用 SDN 技术建立网络的安全运营能力和应急响应机制。例如, 依靠 SDN 细粒度的流量控制能力, 可以方便实现基于流量的异常检测和访问控制, 可用于 DDoS 攻击和企业内网高级可持续威胁 (APT, advanced persistent threat) 攻击的检测和防御。在流量调度方面, SDN 具有灵活编排路由的能力, 可以控制网络流量流经特定安全设备, 从而实现按需保障的安全服务链。

从网络检测方法的角度来看, 在传统网络中, 机器学习被广泛用于流量分类和异常检测^[5]。虽然机器学习方法具有较高的检测精度, 但往往会受到以下几个限制: ① 依赖于海量的、带标签的训练数据, 用于训练分类器^[6]; ② 依赖于大量的

计算资源,用于及时处理和检测恶意行为^[7];③依赖于多样化的网络设备资源,如管理系统和各种网络中间盒,用于收集网络状态信息。此外,基于熵的技术也被用于传统网络中的异常流量检测^[8]。这些技术使用熵理论建模网络状态,通过计算分析一定时间段内的熵值变化,可以检测出熵值的异常偏差以检测网络中何时发生干扰。研究表明,基于熵的异常检测方法是一种易部署、低成本、实时和精确地监测网络安全的技术^[9]。基于熵的入侵检测被广泛研究,然而,这些方法通常应用在大规模的 IP 骨干网络中,对小规模网络或其他类型网络的适用性不强^[8-11]。同时,传统入侵检测方案只涉及异常行为的检测及分类,大部分研究没有关注在检测后如何对网络异常行为进行诊断和防御。

3 相关假设与模型说明

3.1 相关假设

SDN 的优势在于对网络全局流量的检测和控制,便于细粒度地、实时地分析、管理、控制网络,这种网络范式给传统网络安全模型带来冲击,主要体现在以下两方面。

(1) 安全设备部署方式的差异:在传统网络中,入侵检测系统等安全设备被部署在网络的关键位置,要求敏感数据必须从安全设备中通过,安全设备才可以进行实时监控和检测;而在 SDN 中,数据路径均由 SDN 控制器控制,安全设备的位置约束被解除。

(2) 安全状态判定的方式不同:在传统网络中,网络管理员必须向多个安全设备发送状态请求信息,收到信息后,进行综合评估,才能得到网络安全态势;而在 SDN 中,控制器通过定期向数据平面采集信息,从而掌握网络全局态势。

针对 SDN 特点,本文提出一套面向 SDN 的自主防护系统,系统作为 SDN 应用运行于 SDN 控制器之上。当前,OpenFlow 协议已经成为 SDN 的主流南向接口协议,因此系统选择 OpenFlow 流表信息作为异常行为检测的数据源,所以需要 SDN 控制器和网络交换机支持 OpenFlow 协议。

3.2 攻击模型说明

本文使用一个僵尸网络的嗅探、传播及发动

DDoS 攻击的 3 个阶段作为攻击模型,这 3 个阶段包含以下网络异常行为。

(1) 端口扫描(port scanning)。攻击者通常使用端口扫描来嗅探远程主机中的一组端口是否开启。某些开放端口可被攻击者利用漏洞,以攻击目标系统或进行蠕虫病毒传播(worm propagation)^[12]。通常,攻击者使用攻击脚本对要攻击的服务器发起端口扫描,脚本向 0 到 65536 之间的端口发送多个 TCP 连接数据包。当发现一个开放端口时,脚本会返回给攻击者一个通知,攻击者由此判断该端口可以做何种类型攻击。通常,端口扫描在网络中生成具有固定 IP 地址但 TCP 目的端口号不同的数据包。

(2) 蠕虫病毒传播。蠕虫病毒是一段可以自我复制的程序,试图通过利用特定的漏洞感染其他机器。僵尸网络主要通过主动漏洞攻击、即时通信软件、邮件病毒、恶意网站脚本、特洛伊木马等方式传播^[13]。攻击者还会结合蠕虫技术改进僵尸程序传播方式^[14],从而使僵尸程序能够进行自动传播,如著名的僵尸程序样本 AgoBot^[15]。采用蠕虫技术传播的僵尸程序能够快速构建大规模的僵尸网络。

(3) 分布式拒绝服务攻击。一旦 SDN 中形成一定规模的僵尸网络,即可发动 DDoS 攻击^[16]。DDoS 攻击方式有多种,可通过制造大量虚假流量,从而使特定交换机的流表溢出,或通过生成多条 packet_in 消息使 SDN 控制器过载。通常,这些攻击表现为大量不同的源 IP 地址访问某一个目的 IP。本文使用了半开式连接(TCP-SYN flood)攻击,即攻击者选择一个服务器,向该服务器提供的服务端口发送多个 TCP SYN 包。假设有服务在这些端口中运行,服务器将处理一个请求,分配资源来处理它,并向主机请求发送一个 ACK。同时,攻击者可以使用源 IP 地址欺骗,即用另一台机器的源 IP 地址发送 TCP SYN 数据包,导致这些主机错误地收到未经请求的 ACK,使被攻击服务器始终保持开放的 TCP 连接,从而耗尽服务器资源。

本文采用以上 3 种网络异常行为作为攻击模型,主要目的是验证所提方法的有效性。基于 Renyi 熵的检测方法不限于以上网络异常行为,只要网络异常能够在 OpenFlow 流表相应特征中

引起熵值突变,就可以在本方法中被检测出。鉴于此,本文在检测模块设计时引入了“恶意为标记向量”的概念,便于后续工作的扩展。

4 系统设计

4.1 基本概念

(1) 香农熵与 Renyi 熵

熵测量了与概率分布 X 相关的事件的不确定性。对于给定的离散变量 X ,可能出现的结果为 $\{x_1, x_2, \dots, x_n\}$,那么香农熵的形式化定义为

$$H(X) = \sum_{i=1}^n p_i \lg \frac{1}{p_i} = - \sum_{i=1}^n p_i \lg p_i \quad (1)$$

其中, $p_i \in \{p_1, p_2, \dots, p_n\}$, p_i 是随机变量 x_i 的概率分布。熵值越高,说明变量 X 随机性越强;熵值越低,变量 X 确定度越高。

香农熵在各个领域都有大量应用,但针对小流量的度量,Renyi 熵比香农熵更能增大不同分布的随机变量之间的差异。Renyi 熵定义如式(2)所示。

$$H_\alpha(X) = \frac{1}{1-\alpha} \lg \left(\sum_{i=1}^n p_i^\alpha \right) \quad (2)$$

其中, $\alpha \geq 0$, 且 $\alpha \neq 1$ 。可以证明,当 $\alpha=0$, $H_\alpha(X)$ 取得最大值,即

$$\max(H_\alpha(X)) = \lg n \quad (3)$$

当 $\alpha \rightarrow 1$ 时, Renyi 熵收敛于香农熵,即

$$\lim_{\alpha \rightarrow 1} H_\alpha(X) = - \sum_{i=1}^n p_i \lg p_i \quad (4)$$

(2) 网络流

网络流是在一段时间内具有相同特征的网络数据包的集合。本文使用一个包含源 IP 地址、源端口号、目的 IP 地址、目的端口号、源 MAC 地址以及目的 MAC 地址的六元组标识一条流,用 F 表示,则有 $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$ 。六元组中每一位指标称作一位“位元”。用 f_i 表示。进一步地,位元 f_i 对应的第 q 个样本则用 f_i^q 表示。例如,源 IP 地址位元对应的一个具体的 IP 地址。

(3) 流表信息快照

流表信息快照是一个汇总了当前网络中所有流表信息的数据结构,本质上是一个流表项集合,

用 $S = \{F : (c, b, d)\}$ 表示,其中三元组 (c, b, d) 为流 F 的特征集合, c 表示匹配的包数, b 表示传输字节数, d 表示流表项活跃时间。

(4) “位元样本-频数”映射表

“位元样本-频数”映射表用于统计相应位元的样本出现的频数。用 $h_i = \{f_i^1 : m_1, f_i^2 : m_2 \dots f_i^q : m_q, \dots\}$, m_1 代表相应位元 f_i 的样本 f_i^1 出现的频数,如对于第一位位元 f_1 = 源端口号,则映射表 $h_1 = \{f_1^1 : m_1, f_1^2 : m_2 \dots f_1^q : m_q, \dots\}$ 用于记录不同的端口号样本 f_i^q 出现的频数 m_q 。

(5) 恶意为标记向量

以二进制编码的方式标记一种网络恶意为,恶意为对照如表 1 所示。恶意为标记向量是一个对应于流表信息快照中六元组的 6 维向量,记为 $R = \{r_1, r_2, r_3, r_4, r_5, r_6\}$,向量中的每一位 r_i 是一个布尔值,表示六元组中的相应位元是否发生异常。一旦有异常发生,则相应位置为“1”。例如,网络没有异常时,恶意为标记向量为(0,0,0,0,0,0)。然而,当有端口扫描行为发生时,网络中报文的端口号种类会大幅增加,相应的熵值会陡增,异常检测模块一旦检测到熵值变化,则将目的端口号位置“1”。此时,恶意为标记向量为(0,1,0,0,0,0)。

表 1 异常行为对照
Table 1 Abnormal behavior comparison

恶意为标记向量	网络异常行为	异常行为特点
(0,1,0,0,0,0)	端口扫描	目的端口号熵值显著增加
(0,1,1,0,0,0)	蠕虫病毒传播	目的端口号和源 IP 熵值显著下降
(0,1,0,1,0,0)	DDoS 攻击	目的 IP 和目的端口号显著下降

4.2 总体设计

在 SDN 中的异常行为检测、诊断及防御可以通过利用 OpenFlow 交换机流表信息及 OpenFlow 协议行为实现。为了验证这一理念,本文设计并实现了原型系统。本章对原型组件进行详细描述并探讨原型系统。

4.2.1 工作流程

如图 1 所示,系统工作流程包含两个阶段:网络监测阶段与网络诊断阶段。网络检测阶段主

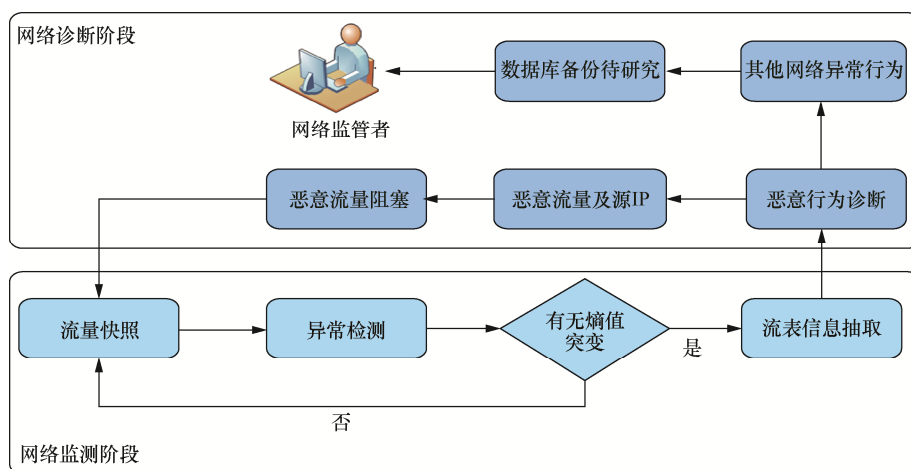


图1 系统工作流程
Figure 1 System workflow

要负责流表信息采集，形成流量快照，之后对流量相应特征进行熵运算，即图1所示的异常检测。一旦发现熵值异常，则抽取相应流量项提交恶意行为诊断模块。网络诊断阶段主要负责恶意行为的诊断及防御。恶意行为诊断模块接收流表项，从中诊断出恶意流量及产生恶意流量的源IP，将其提交网络防御模块，对恶意流量采取阻塞操作。系统可以实现3种恶意行为的判定，对于其他未判明的异常行为提交数据库记录，在后续工作中进一步分析。

4.2.2 系统组件

(1) 信息采集模块：主要负责周期性采集数据平面的流量信息，流量信息以流表信息快照的形式呈现。信息采集模块通过表现层状态转换接口 (Rest API, representational state transfer application interface)，每 n s 向网络控制器发送流表信息提取请求，以查询数据平面中所有流量的状态。从网络控制器接收到流表信息后，信息采集模块解析并组织相关数据，形成一份流表信息快照。本文使用六元组 $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$ 标识一条流。六元组与相应 OpenFlow 匹配域如表2所示。

表2 六元组与相应 OpenFlow 匹配域
Table 2 Six tuples and corresponding OpenFlow matching fields

源端口	目的端口	源 IP	目的 IP	源 MAC	目的 MAC
tcp_src	tcp_dst	ipv4_src	ipv4_dst	eth_src	eth_dst

每一条流对应一个三元的特征集合，特征集

合用于描述这条流的状态。特征集合由 OpenFlow 规范^[3]定义，如表3所示，并表示为一个三元组 (c, b, d) ，其中 c 表示匹配的包数， b 表示传输字节数， d 表示流表项活跃时间。

表3 特征集合信息
Table 3 Features set information

匹配包数	传输字节数	活跃时间
packet_count	byte_count	duration_sec

从而，可以形成一份流表信息快照 $S = \{F : (c, b, d)\}$ 。

(2) 异常检测模块：主要负责根据流量快照信息检测网络中的异常流量。在此模块中，本文主要使用基于 Renyi 熵的检测方法，同时，在后续实验中本文使用传统香农熵和 Renyi 熵做对比。具体做法是：先针对流量快照六元组中每一个位元 f_i ，如目的 IP，分别计算熵值 E_i ；然后，判断 E_i 是否在区间 $[M_i - s_i, M_i + s_i]$ ，若 E_i 为异常值，则将恶意行为标记向量 $R = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ 的相应位元置为“1”，并向网络诊断模块提交恶意行为标记向量和此时的流表信息快照。

(3) 网络诊断模块：接收异常检测模块提交的恶意行为标记向量和发生熵突变的流表信息快照。首先，根据恶意行为标记向量，识别出引起熵突变的位元，判断恶意行为类型（端口扫描、蠕虫病毒传播、DDoS 攻击）。然后，将活跃时间小于 n s 的流表项判定为潜在的恶意流量项。这

一判断基于以下的认知：活跃时间小于 n s 的流表项一定是在这个快照周期内新下发的流量项，而引起熵值突变的流一定在新下发的流表项之中。接着，统计在恶意流表项中源 IP 出现的频率。出现频率最高的 IP 或 IP 集合是产生恶意流量的源 IP 或 IP 集合（若诊断出的恶意行为是 DDoS 攻击，则对应一个源 IP 集合）。相应的流表项代表了恶意流量。最后，形成 JSON 格式的诊断报告，提交网络防御模块。

(4) 网络防御模块：接收网络诊断模块提交的诊断报告，从诊断报告中提取恶意源 IP 和恶意流量对应的流表项。通过 REST API 北向接口指导控制器，下发防火墙规则，将恶意 IP 加入黑名单并阻塞相应恶意流量。

4.3 算法实现

4.3.1 基于 Renyi 熵的异常检测算法实现

本文应用 Renyi 熵理论来检测连续流量快照中某些流量特征分布的变化。例如，考虑在 t_1 和 t_2 两个时刻分别采集到的连续的流表信息快照。如果 t_1 中流量特征的 Renyi 熵近似于 t_2 中相同特征的熵，则认为两个连续快照之间没有发生显著的流量特征变化。但是，如果两个快照之间的流量特征计算的 Renyi 熵存在较大差异，则表示出现了网络异常行为。

基于 Renyi 熵的异常检测如算法 1 所示。检测算法以流表信息快照 S 作为输入，经过检测后，输出恶意行为标记向量 $\mathbf{R} = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ 。首先，遍历六元组 F 中的所有位元 f_i ，为每一个位元 f_i 建立一个“位元样本-频数”映射表 $h_i = \{f_i^1 : m_1, f_i^2 : m_2\}$ ， m_1 代表相应位元 f_i 的样本 f_i^1 出现的频数（此处假设位元 f_i 出现了两种样本 f_i^1 和 f_i^2 ）。然后，继续遍历流表信息快照 S 中的每一条流表项 $\{F_j : (c_j, b_j, d_j)\} \in S$ ，统计流表项中对应位元 f_i 样本 f_i^q 出现的频数 m_q 。遍历完所有表项之后计算所有样本出现的总频数 $\text{Sum}_i = \sum_{q=1}^n m_q$ ，

总频数在求频率时充当分母。然后，再依此遍历映射表 h_i 中每一条样本记录，计算样本对应的频率。最后代入式(2)求对应位元 f_i 的信息熵 E_i 。进

一步地，可按以下方式判定 E_i 是否异常： M_i 表示网络中观察到的平均熵， s_i 表示相关的标准差，如果 E_i 在区间 $[M_i - s_i, M_i + s_i]$ 内，那么 E_i 是正常熵，否则是异常熵；若 E_i 为异常值，则算法将恶意行为标记向量 $\mathbf{R} = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ 的相应位元置为“1”，并返回恶意行为标记向量。

算法 1 基于 Renyi 熵的异常检测算法

输入 流表信息快照 S

输出 恶意行为标记向量 $\vec{\mathbf{R}}$

- 1) for $f_i, f_i \in F$ do
- 2) 建立映射表 $h_i = \{f_i : m\}$
- 3) for $F_j : (C_j, b_j, d_j) (F_j : (C_j, b_j, d_j) \in S)$ do
- 4) 统计位元 f_i 样本出现频数，保存在 h_i 中
- 5) end for
- 6) 计算所有样本出现总频数 Sum_i
- 7) for $f_i^q : m_q \in h$ do
- 8) 计算样本 f_i^q 出现的频率 p_i^q
- 9) end for
- 10) 按照式(2)计算位元 f_i 的 Renyi 熵 E_i
- 11) if $E_i \notin [M_i - s_i, M_i + s_i]$ then
- 12) 将恶意行为标记向量 $\mathbf{R}$ 的第 i 位 r_i 置“1”
- 13) end if
- 14) end for
- 15) return 恶意行为标记向量 $\mathbf{R}$

4.3.2 恶意行为诊断算法实现

恶意行为诊断如算法 2 所示。诊断算法以出现异常时的流表信息快照 S 和恶意行为标记向量 $\mathbf{R} = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ 作为输入，经过诊断后，输出 JSON 格式的诊断报告。诊断报告包括“恶意行为类型”“恶意源 IP 集合”“恶意流集合”。首先根据异常行为向量 $\mathbf{R} = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ ，对照表 1 识别出相应的网络异常行为。然后遍历流表信息快照，将活跃时间小于 n s 的流表项加入潜在的恶意流表项集合 S^p 。然后遍历 S^p ，统计其中源 IP 样本的出现频率，将出现频率最高的源 IP 加入恶意 IP 集合，源 IP 对应的流表项加入恶意流表项集合 S^{ab} 。最后形成 JSON 格式诊断报告并返回。

算法 2 恶意行为诊断算法

输入 流表信息快照 S ，恶意行为标记向量 $\mathbf{R}$

输出 诊断报告

- 1) for $F_j: (C_j, b_j, d_j) // F_j: (C_j, b_j, d_j) \in S$ do
- 2) if $d_j \leq n$ then
- 3) 将流表项 $F_j: (C_j, b_j, d_j)$ 加入集合 S^p
- 4) end if
- 5) end for
- 6) for $F_j: (C_j, b_j, d_j), F_j: (C_j, b_j, d_j) \in S^p$
- 7) 统计源 IP 位元 f_3 的样本出现频率
- 8) end for
- 9) 将出现频率最高的源 IP 样本加入恶意源 IP 集合
- 10) 将此 IP 样本对应的流表项加入恶意流集合 S^{ab}
- 11) 形成诊断报告
- 12) return 诊断报告

4.3.3 网络防御算法实现

网络防御算法以诊断报告为输入，从诊断报告中提取恶意源 IP 和恶意流量，如算法 3 所示。通过北向接口（一般是 Rest API），向控制器下发控制指令指导控制器，下发新的防火墙规则，将恶意 IP 加入黑名单并阻塞相应恶意流量。在每个检测的时间片内查看黑名单，一旦黑名单中的恶意 IP 已无异常行为，则将其移出黑名单。

算法 3 网络防御算法

输入 诊断报告

输出 $\emptyset$

- 1) 从诊断报告中提取恶意 IP 集合
- 2) 将恶意 IP 加入黑名单
- 3) 构造 JSON 格式 Request 命令
- 4) 通过 REST API 下发控制命令阻塞异常流量
- 5) 打印 Response 消息
- 6) return $\emptyset$

4.4 系统时间复杂度分析

本文系统包括 4 个功能组件，其中 3 个功能组件需要处理网络流表项，其时间复杂度应该是网络流表项数量 n 的函数。网络防御模块只采取阻塞动作，所以时间消耗可以忽略。信息采集模块采集所有流表项信息并形成流表信息快照，它必须逐一处理所有流表项，所以其时间复杂度是

$O(n)$ 。异常检测模块依此计算各个位元的 Renyi 熵，位元一共有 6 个。同时每次计算熵时都需要遍历流表信息快照，流表信息快照的规模同流表项规模一致也是 n 。所以，异常检测模块的时间复杂度是 $6O(n)=O(n)$ 。网络诊断模块也需要逐条判断流表项的活跃时间，所有网络诊断模块的时间复杂度也是 $O(n)$ 。因此，整个系统的时间复杂度是 $O(n)+O(n)+O(n)=O(n)$ 。

5 系统实现

本文在 Floodlight 控制器上开发了基于 Python 语言的系统原型。本文系统架构如图 2 所示。Floodlight 是一个支持 OpenFlow 的控制器，它实现了对 OpenFlow 网络的监控和查询功能，同时对上层网络应用提供基于 REST API 的北向接口。本文的原型系统中信息采集模块和网络防御模块需要通过 REST API 接口和 Floodlight 控制器交互。信息采集模块通过 REST API 接口向控制器请求交换机流表信息，网络防御模块通过 REST API 接口调用控制器的访问控制功能，阻塞攻击者产生的恶意流量。

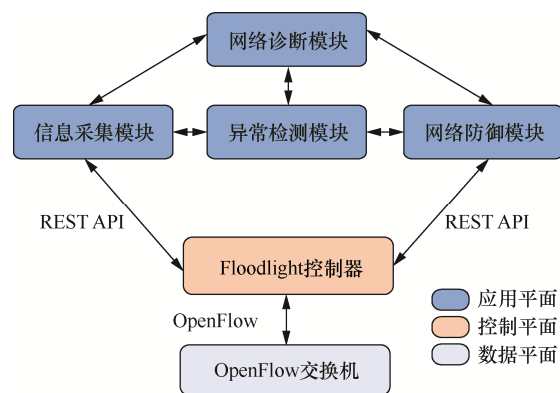


图2 系统架构

Figure 2 System framework

6 性能评估

6.1 实验设置

本文实验环境运行于一台虚拟机上，处理器主频 1.8 GHz，双核，2.5 GB 内存。操作系统为 Ubuntu 14.04 64 位系统。实验基于 Mininet 仿真平台搭建网络拓扑^[17]。此外，实验使用了 Scapy 工具来模拟网络中正常业务流量，并生成攻击流

量^[16]。SDN 采用了 OpenFlow v1.3 版本协议、Floodlight 控制器和 OpenvSwitch 交换机。本实验使用了线性拓扑, 包含一台 Floodlight 控制器、一台 OpenvSwitch 交换机和通过交换机连通的 20 台主机。

6.2 量化评估

实验模拟了一个僵尸网络的嗅探、传播及发动 DDoS 攻击的 3 个阶段中的恶意流量。本文检测了在端口扫描、蠕虫传播以及 DDoS 攻击发生时网络流量的熵值变化情况, 并通过诊断和防御手段抑制攻击。为了可以更清晰地观察到攻击发生时的熵值变化, 本文适当延后了防御动作的启动时间, 从而增加了攻击的持续时间。图 3 中蓝色线条表示香农熵变化情况, 红色线条表示 Renyi 熵变化情况。

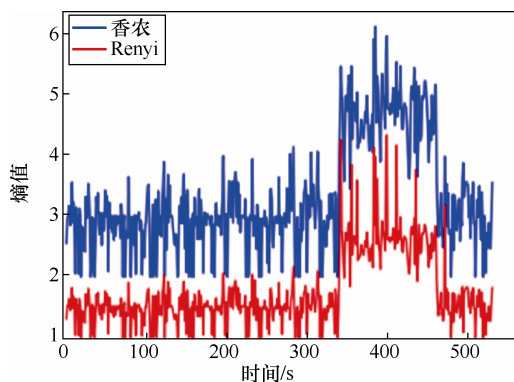


图 3 端口扫描时目的端口号的熵值变化情况
Figure 3 Entropy change of destination port number during port scanning

图 3 给出当有端口扫描行为发生时, 目的端口号字段的熵值变化情况。在 340 s 时开始进行端口扫描, 450 s 开始进行防御。由于攻击者在被攻击主机的目的端口上发动了扫描, 网络中流量的目的端口号种类迅速增多, 这一变化表现从 340 s 开始, 目的端口号熵值陡增。450 s 开始实施防御, 发动端口扫描的流量被交换机丢弃, 因此与端口扫描行为相关的流表项超时失效, 端口号熵值恢复正常。

由图 3 可以发现, Renyi 熵相对于香农熵在正常流量时熵值变化幅度更小, 更易区分正常流量和恶意流量的分布。

图 4、图 5 给出当网络中有蠕虫病毒传播时, 目的端口号字段和源 IP 地址的熵值变化情况。在

100 s 时攻击者开始散播蠕虫病毒, 200 s 时系统开始进行防御。攻击者根据特定目的端口号上存在的漏洞, 发动蠕虫病毒传播, 此时网络中的特定目的端口号上的报文增多, 其他端口上的报文数量相对较少。由于网络报文在特定目的端口号上集中, 所以目的端口号熵值平稳下降。

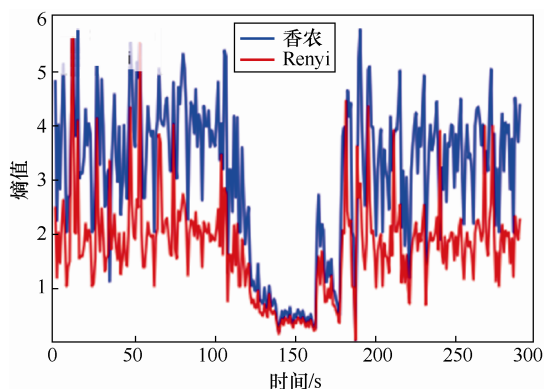


图 4 蠕虫传播时目的端口号的熵值变化情况
Figure 4 Entropy change of destination port number during worm propagation

如图 5 所示, 网络中用于传播蠕虫病毒的流量源 IP 地址均为攻击者的 IP 地址, 所以源 IP 地址的熵值也会下降。但是, 网络中源 IP 地址的种类较多, 随机度较高, 所以即使网络中存在蠕虫病毒传播行为, 源 IP 地址的熵值下降相对要缓和。

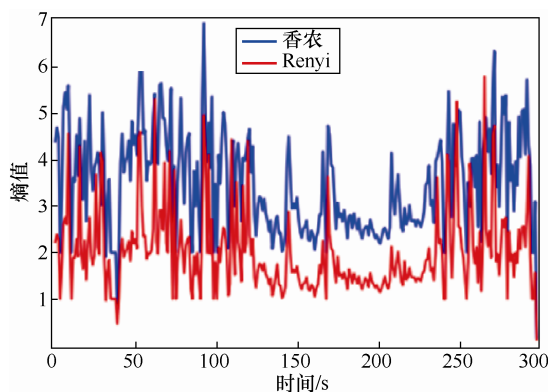


图 5 蠕虫传播时源 IP 的熵值变化情况
Figure 5 Entropy change of source IP during worm propagation

图 6、图 7 分别给出当攻击者发动 DDoS 攻击时, 目的端口号字段和目的 IP 地址的熵值变化情况。攻击者 200 s 时开始散播蠕虫病毒, 系统于 310 s 开始进行防御。当网络中存在 DDoS 攻击时, 网络流量在特定目的端口号上和特定目的 IP 地址上集中, 且 DDoS 攻击强度要远高于蠕虫

病毒传播速度。所以网络中在目的端口号、目的 IP 地址上的熵值陡降。

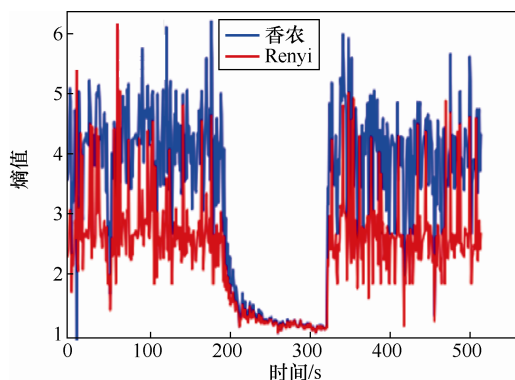


图6 DDoS攻击时目的端口号的熵值变化情况

Figure 6 Entropy change of destination port number in DDoS attack

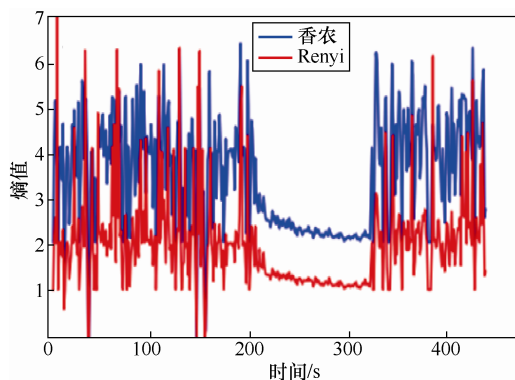


图7 DDoS攻击时目的IP的熵值变化情况

Figure 7 Entropy change of destination IP in DDoS attack

7 结束语

本文提出了一套基于Renyi熵的SDN自主防护系统,该系统利用了SDN集中控制的优势。首先,系统收集OpenFlow交换机流表信息,针对流表项匹配域的不同字段,计算熵值并和阈值进行比较,判断是否有网络异常行为发生。然后,系统通过进一步分析OpenFlow流表的生存时间以及统计IP的出现频率,从而诊断出实施网络异常行为的源IP。最后,系统针对恶意流量实施通信阻塞等防护措施。本文利用Mininet网络仿真工具,搭建了实验环境进行仿真模拟,对比了香农熵和Renyi熵的检测效果,验证了所提出系统的有效性。本文提出的系统特色是直接利用了OpenFlow流表信息,不需要任何第三方网络测量设备,部署代价更低。本文提出的系统可以检测

出3种常见的网络异常行为,后续的研究将结合机器学习技术以及其他检测方法,以进一步丰富可检测的异常行为类别。

参考文献:

- [1] BRAGA R, MOTA E, PASSITO A. Lightweight DDoS flooding attack detection using NOX/OpenFlow[C]//The 35th Annual IEEE Conference on Local Computer Networks. 2010.
- [2] BEHAL S, SINGH J. Detection and mitigation of DDoS attacks in SDN: a comprehensive review, research challenges and future directions[J]. Computer Science Review, 2020, 37: 1-25.
- [3] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2):69-74.
- [4] ZHAO P, ZHAO W, LIU Q. Multi-objective link-separation multipath selection using k max-min for software-defined MANET[C]//2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN). 2019: 253-258.
- [5] SALMAN O, ELHAJJ I H, KAYSSI A, et al. A review on machine learning-based approaches for Internet traffic classification[J]. Annals of Telecommunications, 2020: 1-38.
- [6] WILLIAMS N, ZANDER S, ARMITAGE G. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification[J]. ACM SIGCOMM Computer Communication Review, 2006, 36(5): 5-16.
- [7] LAKHINA A, CROVELLA M, DIOT C. Mining anomalies using traffic feature distributions[C]//ACM Sigcomm Conference on Applications. 2005: 217.
- [8] FIADINO P D, DALCONZO A, SCHIAVONE M, et al. Challenging entropy-based anomaly detection and diagnosis in cellular networks[J]. ACM Sigcomm Computer Communication Review, 2015, 45(4): 87-88.
- [9] TIMCENKO V V, IBRAHIM J, GAJIN S. The hybrid machine learning support for entropy based network traffic anomaly detection[C]//ICIST 2019. 2019: 144-149.
- [10] IBRAHIM J, TIMCENKO V V, GAJIN S. A comprehensive flow-based anomaly detection architecture using entropy calculation and machine learning classification[C]//ICIST 2019. 2019: 138-143.
- [11] SHUKLA A S, MAURYA R. Entropy-based anomaly detection in a network[J]. Wireless Personal Communications, 2018, 99: 1487-1501.
- [12] SCHAEFFER-FILHO A, MAUTHE A, HUTCHISON D, et al. PreSET: a toolset for the evaluation of network resilience strategies[C]//IFIP/IEEE International Symposium on Integrated Network Management. 2013.
- [13] STONE-GROSS B, COVA M, GILBERT B, et al. Analysis of a Botnet Takeover[J]. IEEE Security & Privacy, 2011, 9(1): 64-72.
- [14] WANG Q, CHEN Z, CHEN C, et al. On the robustness of the bot-

net topology formed by worm infection[C]//Global Telecommunications Conference. 2010.

- [15] 周佳骏, 汪婷婷. 基于计算机网络对抗的 Agobot 实例分析[J]. 网络安全技术与应用, 2013, (7): 86-88.

ZHOU J J, WANG T T. Instance analysis of Agobot based on computer network operations[J]. Network Security Technology & Application, 2013, (7): 86-88.

- [16] 李世杰, 倪洪. 面向DDoS防御设备安全性测试的Scapy应用[J]. 网络安全技术与应用, 2020, (1): 20-22.

LI S J, NI H. Scapy application for DDoS defense equipment security test [J]. Network Security Technology and Application, 2020, (1): 20-22.

- [17] KAUR K, SINGH J, GHUMMAN N S. Mininet as Software Defined Networking Testing Platform[C]//International Conference on Communication, Computing & Systems. 2014.

[作者简介]



赵普(1991-), 男, 河南安阳人, 国防科技大学硕士生, 主要研究方向为软件定义网络安全。



赵文涛(1978-), 男, 博士, 内蒙古凉城人, 国防科技大学教授、博士生导师, 主要研究方向为网络性能优化、信息处理和机器学习。



付章杰(1983-), 男, 博士, 河南南阳人, 南京信息工程大学教授、博士生导师, 主要研究方向为网络与信息安全等。



刘强(1986-), 男, 博士, 江西临川人, 国防科技大学副教授, 主要研究方向为网络安全和机器学习。